B6

inserted at bit position N, as shown in Fig. 5. The output of the hash function, together with the inserted color value, is stored in the HashedKey register 106. If colors are disabled for a tree, the 176-bit hash function is taken unmodified, and 16 zeros are appended to the hash output to produce the 192-bit final HashedKey. –

Page 40, amend paragraph beginning at line 8 as follows:

B7

– When a PSCB is encountered during a search in an LPM tree, the tree search engine hardware 70 will continue the tree-walk on the 0-branch or the 1-branch, depending on the value of the bit p of the UnhashedKey. –

In the Claims:

Amend Claims 1 and 35 as follows:

C1
B8

1. (Amended) A method for determining a longest prefix match for a variable length search key by a computer processing device, comprising the acts of:
    - reading an input key as a search string;
    - using the N most significant bits of the input key as an address to index into a table representing a plurality of root nodes of search trees wherein each non-empty entry contains a pointer to a next branch in the search tree or a leaf;
    - determining if the pointer in a non-empty table entry points to a leaf or a next branch of the corresponding search tree;
    - reading the next branch contents if the pointer does not point to the leaf of the corresponding search tree and comparing the prefix represented by the next branch with the input key to find a distinguishing bit

C1
B8

position;

reading a leaf pattern when the leaf of a corresponding search tree is reached and comparing the leaf pattern with the input key to determine if the leaf pattern matches the input key; and

returning the longest prefix match found for the input key to a requesting application.

C1
B9

35. (Amended) A computer readable medium containing a program product for determining a longest prefix match for a variable length search key, comprising:

program instructions that read an input key as a search string;

program instructions that use the N most significant bits of the input key as an address to index into a table representing a plurality of root nodes of search trees wherein each non-empty entry contains a pointer to a next branch in the search tree or a leaf;

program instructions that determine if the pointer in a non-empty table entry points to a leaf or a next branch of the corresponding search tree;

program instructions that read the next branch contents if the pointer does not point to the leaf of the corresponding search tree and compare the prefix represented by the next branch with the input key to find a distinguishing bit position;

program instructions that read a leaf pattern when the leaf of a corresponding search tree is reached and compare the leaf pattern with the input key to determine if the leaf pattern matches the input key; and

program instructions that return the longest prefix match found for the input key to the requesting application.

Add the following new claims 47-59:

C1

B10

47. A method for determining a longest prefix match for a variable length search key by a computer processing device, comprising the acts of:

reading an input key as a search string;

using N bits of the input key as an address to access a table representing a plurality of root nodes of search trees wherein each non-empty entry contains a pointer;

determining if the pointer in a non-empty table entry points to at least a leaf of the corresponding search tree;

reading a leaf pattern in the leaf;

comparing the leaf pattern with the input key, to determine if the leaf pattern matches the input key, only if the leaf is an end leaf; and

returning the leaf pattern as the longest prefix match found for the input key to a requesting application only if there is an exact match between the leaf pattern and the input key.

48. The method of claim 47 further including the acts of:

storing the leaf pattern in a stack if the leaf is not the end leaf of the corresponding tree;

determining other leaves, if any, that are not end leaves of the corresponding tree; and

storing leaf patterns associated with each one of the other leaves in the stack.

49. The method of claim 48 further including the acts of retracing a path laid out between the end leaf, other leaves and root node of the corresponding tree to

determine the longest prefix match if the leaf pattern in the end leaf does not exactly match the input key.

C1
B10

50. The method of claim 49 wherein the retracing act includes correlating the input key with the leaf pattern in the end leaf to determine a distinguishing bit position;

using the distinguishing bit position to access the stack; and

selecting one of the entries thereon as the longest prefix match found for said input key.

51. A method for determining a longest prefix match for a variable length search key by a computer processing device, comprising the acts of:

(a) reading an input key as a search string;

(b) using the N most significant bits of the input key as an address to index into a table representing a plurality of root nodes of search trees wherein each non-empty entry contains a pointer to a next branch in the search tree or a leaf;

(c) determining if the pointer in a non-empty table entry points to a leaf or a next branch of the corresponding search tree;

(d) reading the next branch contents if the pointer does not point to a leaf of the corresponding search tree;

(e) repeating acts (c) and (d) until a leaf is reached;

reading a leaf pattern when the leaf of a corresponding search tree is reached and comparing the leaf pattern with the input key to determine if the leaf pattern matches the input key; and

returning the leaf pattern as the longest prefix match found for the input key to a requesting application.

C1

B10

52. A data structure for use in database comprising:

(a) a table having a plurality of entries wherein at least one entry includes a first field in which a first leaf bit pattern is stored, a second field in which a bit value indicating the Next Bit to Test (NBT) is stored and a third field in which a Next Pointer Address (NPA) is stored; and

(b) at least one Pattern Search Control Block (PSCB) operatively coupled to the at least one entry, said PSCB including two entries with each entry being structured with fields identical to those listed in (a).

53. The data structure of claim 52 wherein the entries in the PSCB are stacked.

54. The data structure of claim 52 wherein the fields in (a) are serially concatenated forming a line.

55. The data structure of claim 52 wherein the bit value points to a bit in a data stream that is being correlated against the database.

56. The data structure of claim 55 wherein the value of the bit in the data stream determines which one of the two entries to follow in correlating the data stream with the database.

57. A method to correlate a search string against a database comprising the acts of:

(a) providing the database having data structure including a table with N entries N > 1, at least one entry including a first field in which a first leaf bit pattern is stored, a second field in which a bit value is stored and a third field in which a Next Pointer Address (NPA) is stored; and

(b) at least one Pattern Search Control Block (PSCB) operatively coupled to the

C1

B16

at least one entry, said PSCB including two entries with each entry being structured with fields substantially similar to those listed in (a);

using M bits of the search string as an address to index into the at least one entry; and

reading the entry to determine what action next to pursue in correlating the search string with the database.

58. The method of claim 57 further including the acts of if the entry only points to an end leaf, read the pattern recorded in the end leaf;

comparing the pattern with the search string; and

returning the pattern to a requesting application as the longest prefix match found.

59. The method of claim 57 further including the acts of if the entry points to a first leaf and a first branch,

recording the pattern stored in the first leaf in a stack;

reading the information stored in the first branch;

if the information stored in the first branch points to another leaf and another branch, recording the pattern stored in the another leaf in the stack;

reading the information stored in the another branch;

repeating the preceding two acts until the last leaf in the corresponding tree is reached;

comparing the pattern recorded in the last leaf with the search string to find a distinguishing bit position;

using the distinguishing bit position to access the stack; and

selecting one of the entries thereon as the longest prefix match found for said input key.